Virtual Environment for Next Generation Sequencing Analysis

Olivier Terzo, Lorenzo Mossucca
Infrastructure and Systems for Advanced Computing (IS4AC)
Istituto Superiore Mario Boella (ISMB)
Torino, Italy
Email: (terzo,mossucca)@ismb.it

Andrea Acquaviva, Francesco Abate, Rosalba Provenzano
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
Email: (andrea.acquaviva,francesco.abate)@polito.it
rosalba.provenzano@studenti.polito.it

***Abstract*—Next Generation Sequencing technology, on the one hand, allows a more accurate analysis, and, on the other hand, increases the amount of data to process. A new protocol for sequencing the messenger RNA in a cell, known as RNA-Seq, generates millions of short sequence fragments in a single run. These fragments, or reads, can be used to measure levels of gene expression and to identify novel splice variants of genes. The proposed solution is a distributed architecture consisting of a Grid Environment and a Virtual Grid Environment, in order to reduce processing time by making the system scalable and flexible.**

***Keywords*-grid computing; cloud computing; virtual; next generation sequencing; hybrid architecture.**

I. Introduction

Next Generation Sequencing (NGS) technologies, also known as second generation, have revolutionized research in the field of biology and genomic with the ability to draw from a single experiment a larger amount of data sequence with the previous technology known as Sanger Sequencing [1] and [2]. The main novelty introduced by the NGS platform is to obtain from the molecules of DNA/RNA of smaller fragments, called read, which are sequenced in parallel thus reducing the processing time. Aberrant mutations in the RNA transcription, as chimeric transcripts, are on the base of various forms of disease and NGS proved to be extremely helpful in making the detection of these events more accurate and reliable. However, even if from the biological point of view NGS technology leads to new exciting perspectives spreading an incredible amount of data, on the other hand it raised new challenges in the development of tools and informative infrastructures. An NGS machine produces millions of reads in a single run that must be successively elaborated and analyzed. TopHat is a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions. It is built on the ultrafast short read mapping program Bowtie [5]. TopHat finds splice junctions without a reference annotation. By first mapping RNA-Seq reads to the genome, TopHat identifies potential exons, since many RNA-Seq reads will contiguously align to the genome. Using this initial mapping, TopHat builds a database of possible splice junctions, and then maps the reads against this junction to confirm them. The goal is to offer to biologist private infrastructures to conduct their research and to respond to the ever evolving needs of NGS users. Cloud Computing is rapidly emerging as an alternative platform for the computational and data needs of our community. Biologists are already using the Amazon Elastic Cloud Computing (EC2) infrastructure for their research. In some situations, it is preferable to use a number of instances of a tailored Virtual Machine (VM) than submitting jobs to the own existing infrastructure. Grids appear mainly in high performance computing environments. In this context, several of off-the-shelf nodes can be linked together and work in parallel to solve problems, that, previously, could be addressed sequentially or by using supercomputers. Grid Computing is a technique developed to elaborate enormous amounts of data and enables large-scale resource sharing to solve problem by exploiting distributed scenarios. The main advantage of Grid is due to parallel computing, indeed if a problem can be split in smaller tasks, that can be executed independently, its solution calculation speed up considerably.

The paper is organized as follows: Section 2 the motivation is discussed. Section 3 explains the biological background and software used. Section 4 shows the architecture design: grid and virtual environment and schedulers functionalities. Section 5 is related to the test performances. The last section draws the conclusion and direction for future work.

II. Motivation

The amount of data produced with NGS technology is a positive factor that on a hand contributes to make studies more accurate and reliable identification of mutations in aberrant splicing events, fused genes, on the other hand, open new challenges in the development tools and infrastructure that are able to do the post-processing of data produced in a powerful and timely fashion [3]. A NGS data sample consists of millions of reads, and in a classic situation, with only one workstation available, the time needed to obtain the output increases significantly. In such a context, this computing infrastructure allows to improve overall system performance optimizing the use of resources and increasing

 ISBN: 978-1-61208-186-1

the system scalability. Recall also that the alignment is a process in which each mapping reference is made to read independently from the other reads, and this means that you can perform a parallel analysis of the data. Even if the alignment is a very basic operation, due to the great number of data involved in the process, the computational effort in this phase is very high. This scenario recalls for the need of developing computing infrastructures presenting high performances CPU capability and memory availability.

III. Next Generations Sequencing

A. TopHat algorithm

TopHat [7] is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons. TopHat is a collaborative effort between the University of Maryland Center for Bioinformatics and Computational Biology and the University of California, Berkeley Departments of Mathematics and Molecular and Cell Biology. TopHat receives as input reads produced by the Illumina Genome Analyzer, although users have been successful in using TopHat with reads from other technologies. The input samples consist of two files of about 37 million of reads each. The two files are FASTA formatted paired-end reads. Dealing with paired-end reads means that the reads are sequenced by the sequencing machine only on the end of the same DNA/RNA molecule, thus the sequence in the middle part is unknown. Each sequenced end of the same read is also referred as mate. It results in two distinct files, the first one consists in the first mate of the same reads and the second one consists in the opposite mate. TopHat finds junctions by mapping reads to the reference in two phases. In the first phase, the pipeline maps all reads to the reference genome using Bowtie. All reads that do not map to the genome are set aside as initially unmapped reads. Bowtie reports, for each read, one or more alignment containing no more than a few mismatches in the 5'-most s bases of the read. The remaining portion of the read on the 3' end may have additional mismatches, provided that the Phred-quality-weighted Hamming distance is less than a specified threshold. TopHat allows Bowtie to report more than one alignment for a read, and suppresses all alignments for reads that have more than this number. This policy allows so called multireads from genes with multiple copies to be reported, but excludes alignments to low-complexity sequence, to which failed reads often align and then assembles the mapped reads. TopHat extracts the sequences for the resulting islands of contiguous sequence from the sparse consensus, inferring them to be putative exons. TopHat produces a compact consensus file containing called bases and the corresponding reference bases in order to generate the island sequences. TopHat uses the reference genome to call the base. Because most reads covering the ends of exons will also span splice junctions, the ends of exons in the pseudoconsensus will initially be covered by few reads, and as a result, an exons pseudoconsensus will likely be missing a small amount of sequence on each end. In order to capture this sequence along with donor and acceptor sites from flanking introns, TopHat includes a small amount of flanking sequence from the reference on both sides of each island. TopHat has a number of parameters and options, and their default values are tuned for processing mammalian RNA-Seq reads also it can be used for another class of organism.

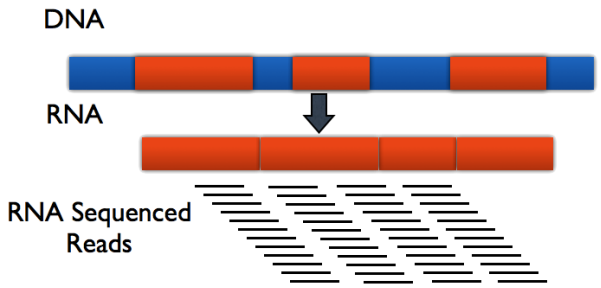

Figure 1. Alignment Phase.

B. Alignment Tools: Bowtie

The short reads alignment is surely the most common operation in RNA-Seq data analysis. The purpose of the alignment is to map each short read fragment onto a genome reference (see Figure 1). From the computational point of view, each short read consists in a sequence of four possible characters corresponding to the DNA bases and the sequence length depends on the sequencing machine adopted for the biological experiment [6]. The main novelty introduced by NGS technology is the capability of sequencing small DNA/RNA fragments in parallel, increasing the throughput and producing very short reads as output. However, this feature make the computational problem more challenging because of the higher amount of read produced and the accuracy in the mapping (the shorter the sequence length, the higher the probability of having multiple matches). For this reason many alignment tools specifically focussed on the alignment of short reads have been recently developed. In the present work, we are interested in characterizing the performances of alignment tools on real NGS data. On the wave of this remark, Bowtie has been chosen, a wide diffused alignment program particularly aimed at align short reads. In order to detect the actual limitation of the alignment phase, we considered real NGS data coming from the analysis of Chronic Myeloid Leukemia. In our analysis flow, the HG19 assembly produced in the 2007 is considered as reference genome the last human genome assembly produced to now. In order to increase the computational performances during

 ISBN: 978-1-61208-186-1

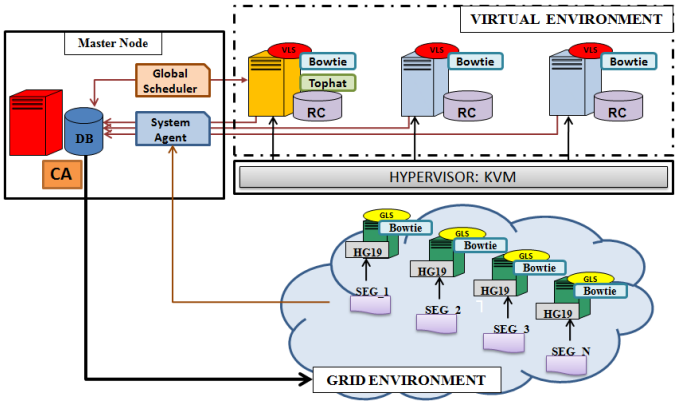

Figure 2. VirtualBio Architecture.

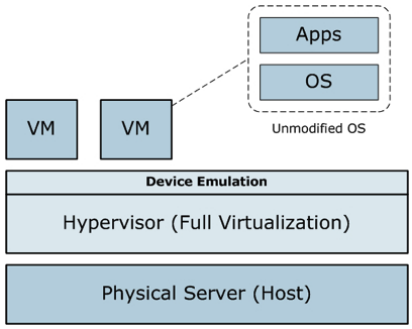

Figure 3. Full Virtualizzation Approach.

the read mapping, Bowtie program creates an index of the provided human genome reference. This operation is particularly straightforward from the computational point of view, but it must be performed only one time for the human genome reference and it is independent on the mapping samples. The alignment phase itself is particularly suitable to be parallelized. In fact, each mapping operation is applied to each read independently on the other read mapping.

C. Distributed Approach

In a preliminary phase of reverse engineering, studying TopHat, blocks of transactions have been highlighted that were executed sequentially. We identified 3 main blocks, that can be executed independently: (a) left and right mate mapped with HG19, (b) segments mapped with HG19, (c) segments mapped with segment juncs. A feature of these 3 blocks is that they are performed by a external software, called Bowtie, as explained before. In steps (a) and (c), since the files involved in the development are significant, we created a common repository that contains the temporary folder used by TopHat. Although the use of a common repository is slightly increased processing time, due to the SSH protocol connection, this time is less than the time of transfer of the entire set of files. Instead the step (b) uses small files these can be performed on a grid, both physical and virtual, because the transfer times are lower. Only difference that the input files are transferred to worker node through Globus Toolkit. These worker nodes when the process is terminated, re-send the output file to the node that requested execution.

IV. VirtualBio Infrastructure

The proposed architecture allows to manage RNA data, prepared by the version of TopHat in Grid, but not only, it could also handle other processing flows that use software and other tools e.g., original version of TopHat or only Bowtie (see Figure 2). The architecture, called VirtualBio, is composed of three main components: a Master Node (MN), a part consists of the Physical Worker Nodes (PWN) that set the grid environment while a part consists of Virtual Worker Nodes (VWN) that set the virtualized environment [9]. The MN is a physical machine with quite good hardware characteristics, is responsible for CA, contains the database, where all information about the nodes belonging to the infrastructure have stored, the node status, the flow of the various biological analysis that can be made in the system and system monitoring. Moreover, on it has been configured the common repository, using the Network File System (NFS). NFS is a protocol developed by Sun Microsystems in 1984 to allow computers to share files and folders over a network [13]. NFS is an open standard, defined in RFCs, and allows any to implement the protocol. Both environments are configured with the middleware Globus Toolkit [11], since it allows obtaining a reliable information technology infrastructure that enables the integrated, collaborative use of computers, networks and databases. The Globus Toolkit is a collection of software components designed to support the development of applications for high performance distributed computing environments, or computational grids.

A. Grid Environment

The Grid environment consists of machines with high computing power, this allows to use own machines and also machines belonging to different virtual organization. The only requirement is to have the necessary software installed for the processing (Bowtie, TopHat and Globus Toolkit). On each worker node of the grid environment is installed the Grid Local Scheduler, an essential component for performing biological tests.

1) Grid Local Scheduler: The Local Grid Scheduler (GLS) is a scheduler active physical machines, has been developed for the design phase (b), it aligns the segment with respect to the human genome (HG19) through Bowtie. Since the transfer of the input file is not influential, the worker nodes do not need to be in the same subnet as the Master Node, but may also belong to different virtual organization, so system can have greater scalability and can use machines powerful performance [10].

 ISBN: 978-1-61208-186-1

Reads	1 CPU	2 CPU	3 CPU	4 CPU	5 CPU	6 CPU	7 CPU	8 CPU
100	31	24	24	24	24	25	32	26
1000	26	38	24	24	24	25	13	20
10000	24	24	22	23	22	23	22	22
100000	28	25	25	24	24	24	24	24
1 E06	62	49	50	42	41	41	38	38
10 E06	424	258	230	185	176	165	158	154
85 E06	3425	2044	1791	1432	1342	1247	1180	1048

Table I
BOWTIE EXECUTION TIME (SECONDS)

B. Virtual Environment

Virtualized environment also helps to improve infrastructure management, allowing the use of virtual node template to create virtual nodes in a short time, speeding up the integration of new nodes on the grid and, therefore, improving the reactivity and the scalability of the infrastructure. The open source KVM has been used as hypervisor. It allows to create Fully Virtualized machines. The kernel component of KVM is included in mainline Linux [12]. KVM allows a Full Virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). KVM is implemented as a module within the Linux kernel. A hypervisor hosts the virtual machine images as regular Linux processes, so that each virtual machine image can use all of the features of the Linux kernel, including hardware, security, storage and applications. Full Virtualization provides emulation of the underlying platform on which a guest operating system and application set run without modifications and unaware that the platform is virtualized (see Figure 3). It implies that every platform device is emulated with enough details to permit the guest OS to manipulate them at their native level. Moreover, it allows administrators to create guests that use different operating systems. These guests have no knowledge about the host OS since they are not aware that the hardware they see is not real but emulated. The guests, however, require real computing resources from the host, so they use a hypervisor to coordinate instructions to the CPU. The main advantage of this paradigm concerns the ability to run virtual machines on all popular operating systems without requiring them to be modified since the emulated hardware is completely transparent. The virtualized environment has pre-installed images, which contain all software and libraries needed for running Bowtie and TopHat. The pre-configured images allow an ease instantiation of the machines when needed, and can be easily shutdown after the use.

1) Virtual Local Scheduler: The Virtual Local Scheduler (VLS) is a scheduler active virtual machines. Its purpose is to draw up the steps (a) and (c) of TopHat. As the GLS, the VLS performs the mapping files for input received through Bowtie. The step (a) allows the alignment with respect to the human genome (HG19) and step (c) allows the alignment with respect to the segment juncs previously constituted by TopHat. Since the considerable size of the files involved in these two steps, the VLS works directly on the temporary folder that is located in the common repository, allowing to avoid wasting time due to the transfer of data. Even in this case the interaction with the database is essential and very frequent, network problems may affect the entire biological analysis.

V. PERFORMANCE CONSIDERATIONS

In a preliminary work, we have introduced two case studies, based on Bowtie execution, from two different points of view. The first is the fragment size, while the other one is the CPUs number on the worker node. In Table I a summary of the calculations obtained changing CPUs number are presented. We can notice for reads between 100 and 1000000 no gain of time has occurred, so for our studies only reads from 1000000 to 85000000 are considered. This is because the files have limited data, thus the processing times are already reduced at this stage and then having multiple processors is irrelevant. As we explained before, during an analysis phase of the algorithm, 3 main blocks have been identified, (a) left and right mate aligned with HG19, (b) segments aligned with HG19, (c) segments aligned with segment juncs. The processing time of each segment depends on parameter pthread that is specified in command of Bowtie and refers to the number of parallel processes that can run. In Figure 4, processing time of a single segment of the variation of the parameter pthread is depicted. The test was run on a machine with the following hardware characteristics: Intel Xeon CPU X5660 @ 2.80 GHz, 12 CPUs and 20 GB of RAM, it is worth noting that in order to gain the maximum time the number of pthread must be equal to the number of CPUs. Once past this threshold, the trend is no longer regular, this is due to the scheduling allocation of the CPU operating system. This test allowed to have a vision on the processing time will have access to machines with different power and CPUs number, opening to a more accurate scheduling policy adapted to the needs of time of the biologist. The Table II instead, depicts the processing time of entire flow of TopHat, comparing the original algorithm (sequential version) with the modified algorithm (parallel version). We obtain a considerable gain

 ISBN: 978-1-61208-186-1

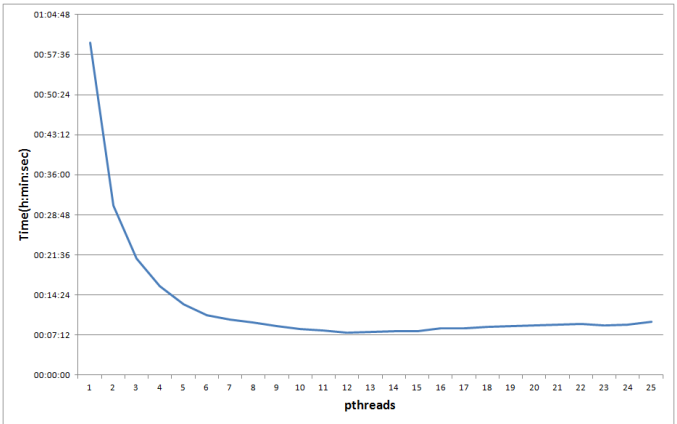

Figure 4. Bowtie Execution Time.

of time that varies depending on the power of machines available. In our tests in order to make homogenous system, we have used machines with the same hardware (12 CPUs).

TopHat	Time [h:min:s]
Sequential Version	03:47:00
Parallel Version	02:10:00

Table II
TOPHAT EXECUTION TIME.

VI. CONCLUSIONS AND FUTURE WORKS

VirtualBio is a tool for NGS analysis, in particular for the alignment through TopHat and Bowtie. The tool wants to offer to biologist private infrastructure to conduct their studies. The novelty of this solution covered both the field of infrastructure and the optimization algorithm of TopHat. Infrastructure is based of two environments: grid and virtual, using a common repository and a set of job schedulers. The TopHat algorithm has been optimized making parallel some sections that were sequential. The architecture allows to reduces the elaboration time by at least 40%. Future work includes the optimization of scheduling policies that are also open to a scenario multisample and implementation of the architecture in Cloud environment, thus increasing the system scalability.

REFERENCES

[1] De Magalhes J.P., Finch C.E. and Janssens G., *Next-generation sequencing in aging research: Emerging applications, problems, pitfalls and possible solutions.*, Ageing Research Reviews, 2010 Jul; Vol. 9(3), pp. 315-323

[2] Sanger F., Nicklen S. and Coulson A.R., *DNA sequencing with chain-terminating inhibitors*, Proc. Natl. Acad. Sci. USA 74, 1977), pp. 5463-5467

[3] Kircher M. and Kelso J., *High-throughput DNA sequencing concepts and limitations.*, Bioessays. 2010 Jun, Vol. 32(6), pp. 524-356

[4] Maher C.A., Palanisamy N., Brenner J.C., Cao X., Kalyana-Sundaram S., Luo S, Khrebtukova I., Barrette T.R., Grasso C., Yu J., Lonigro R.J., Schroth G., Kumar-Sinha C., Chinnaiyan Y., *Chimeric transcript discovery by paired-end transcriptome sequencing*, AM. Proc Natl Acad Sci USA, 2009 July, Vol. 28

[5] Pop M. and S.L. Salzberg, *Bioinformatics challenges of new sequencing technology*, Trends Genet. Vol. 24, 2008

[6] Langmead B., Trapnell C., Pop M. and Salzberg Steven L. *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome*. Genome Biology 10:R25

[7] Trapnell C., Pachter L. and Salzberg Steven L. *TopHat: discovering splice junctions with RNA-Seq*, Bioinformatics (2009), Vol. 25, doi:10.1093/bioinformatics/btp120, pp. 1105-1111

[8] Langmead B., Hansen K. and Leek J. *Cloud-scale RNA-sequencing differential expression analysis with Myrna* Genome Biology (2010) 11:R83

[9] Berman F., Fox G. and Hey A.J.G. *Grid Computing Making the Global Infrastructure a Reality*, Wiley, 2005

[10] Kurowski K., Nabrzyski J., Oleksiak A. and Weglarz J. *Scheduling jobs on the Grid-Multicriteria approach*, Computational Methods in Science and Technology, 12(2), pp. 123-138, 2006

[11] Globus Toolkit, http://www.globus.org/toolkit/, January, 2012

[12] Kernel-based Virtual Machine, http://www.linux-kvm.org/, December, 2011

[13] Network File System, http://wiki.ubuntu-it.org/Server/Nfs, December, 2011

 ISBN: 978-1-61208-186-1